Discovering browsing paths on the web

B. Crémilleux, M. Gaio, J. Madelaine, K. Zreik

GREYC, CNRS - UMR 6072
Université de Caen
F-14032 Caen Cedex France
{cremilleux,gaio,madelaine,zreik}@info.unicaen.fr

Abstract

This paper addresses the problem of being able not only to retrieve documents about a given subject or matching a given description, but to allow a browsing across clusters of documents. This work in progress is based on a first experience of an authoring tool described in the first part of the article.

1 Introduction

The non stop increasing growth of information sources available on the World Wide Web, implies the development of efficient tools to find, extract, filter, and evaluate the needed information and resources. Information sources on the web, such as hypertext documents, are very heterogeneous and weakly structured. It reduces the efficiency of most automated data management system of Web-based information. Thus, the Web becomes the focus of many researches and developments on data mining and machine learning.

Most of these projects are suggesting methods to analyze and discover useful information from the Web, they are often referred as to Web Mining tools (WM) [PA97]. We can distinguish two kinds of WM: the Web Content Mining and the Web Usage Mining. The WCM, is the process of information discovery from sources across on-line database or the Web [SPE97] [CH97] [WVS96]. The WUM, is the process of mining user access patterns (data logs) across the World Wide Web or the on-line databases [CMS97]. In this paper we present some research issues, techniques, and development related to the Web content mining. Our goal is to retrieve single documents and moreover to allow discovery of paths across hyper-documents.

In the Section 2, we will present the design of an authoring tool ("OhRAGE") that allow the building of a consistent hyper-document from single documents. The consistency of the hyper-document is the ability for a end user to browse easily across the document.

Section 3 describes the advantages of making clusters from the Web pages. On the one hand, they provide the ability to construct a hyper-document from pages already designed. On the other hand, they allow the user to build browsing paths closer to his own interests.

2 Hyper-Document construction

2.1 Motivations

Since the Open Source software development we know the two models : cathedral and bazaar. Many companies have a website built with something similar to the cathedral model. They impose a strict graphical chart, and every published page is subject to the approval of the hierarchy or at least to be reviewed by a web editing committee. This model leads to a more or less homogeneous website; however, it does not allow the efficient use of search engines on the site, because if the graphical chart enforces a look and the hierarchy approval enforces a "right" content, nothing is generally done to qualify semantically the content of the pages. We can even say that for basic search engines the whole Web appears as a collection of single bricks with rather no relation to each others.

We will describe below an authoring tool (OhRAGE) that has been designed to build hyper document from single pages. It has mainly been experimented to help users to browse through the global hyper document. It has also been used to test various indexing methods.

2.2 Insertion of a document

We start from the observation that it is difficult for the authors of a Web page to specify sound and accurate keywords for his document. In particular, this remark applies to the keywords that can be precised in the META subelement of the HEAD element of a HTML document. It is not the definition of the Dublin Core completed by the RDF initiative that makes this task easier. Indeed, these initiatives are made to ease the processing of the keywords but not to help authors to choose them.

Even if we can built a search engine that points some documents of our interest, it is difficult for such a tool to propose a way of browsing through these documents. This becomes a challenge with the deployment of Xlink technology [1] that allow to specify a navigation path from outside the basic documents. It will be possible to put navigation orders outside basic documents and to built different browsing paths for the same sets or subsets of pages. This technology allows to link documents with information resources without the need to change these resources that are being linked. There is no requirement to insert informations concerning the links within the content of a document. The links are not in the original document any more. In order to give this functionality to users having a standard HTML browser, today, we can find tools based on XLST processor than can generated pure HTML that represents Xlink elements.

We did not wait for this clean technology to test the ideas of building a consistent hyper documents from single documents. The prototype OhRAGE is an authoring tool. It has been tested for 4 years to build different atlases made of pages written by different geographers. OhRAGE identifies three classes of actors :

editors who have the final responsibility of the global document; they also propose the skeleton of the tree structure consisting of themes (or topics)

[1] see http://www.w3.org/XML/Linking

and subthemes ;

authors who insert new single documents;

readers who can browse the resulting hyper document.

The tool OhRAGE allows an author to insert his document [2]. The system asks him first to specify (through an intuitive graphical interface) the theme, subtheme, sub-subtheme, etc. which it belongs to. Each of these terminal sub-subtheme can be thought as a bag where new contributions are added. This attachment to a particular leaf of the tree skeleton allow a sequential browsing of all the global document through the structural links (see Fig. 1).

In order to allow some random access, the author is first asked to define some "Entrance Descriptors" (ED) that describes what the contribution is talking about. In order to help him in his choice, a list of the descriptors already chosen for other documents is proposed. Each descriptor is either a single keyword or a phrase. This list can be enhanced with descriptors predefined by the editorial staff. Moreover, authors are always allowed to type in their own phrase. These EDs will be used by the restitution system.

Secondly, the author must defined some references. The source of a reference is the part of a text (a word or a phrase), an image, or the portion of an image. The source is then qualified by a list of descriptors, the so called "eXit Descriptors" (XD), which define targets that have to be reached by this reference. The XDs are defined with just the same procedure than EDs. These references will be used during the reading process to build links between documents. The benefit of describing what the targets should be like instead of pointing directly to one or few targets, is that we can virtually point to non-existing document at the time of the insertion.

Lastly, if needed, the author can upload the graphics illustrations or images than can come separately from the text.

2.3 Reading paths across the document

There are two kinds of reading path permitted by restitution system: sequential access to the leaf document through the structural links, and random access via the references. The first way of browsing from one page to a next one is rather classical. The second one needs intermediate pages made of links pointing to the real target pages; this last set of pages is computed with the crossing of their EDs with the XDs of the source reference. The matching criteria we used is rather crude: it is simply the cardinal of the intersection of these two sets. Our experiments show that this method has satisfying results when we deal with documents of the same domain.

2.4 Experiments

This tool has been used to create online versions of three geographical atlases, constituted with a few hundreds of documents. We tested the soundness of the EDs by trying to augment these descriptors with the most frequent word to be found inside a given document.

[2] Here, a document means a hTML page.

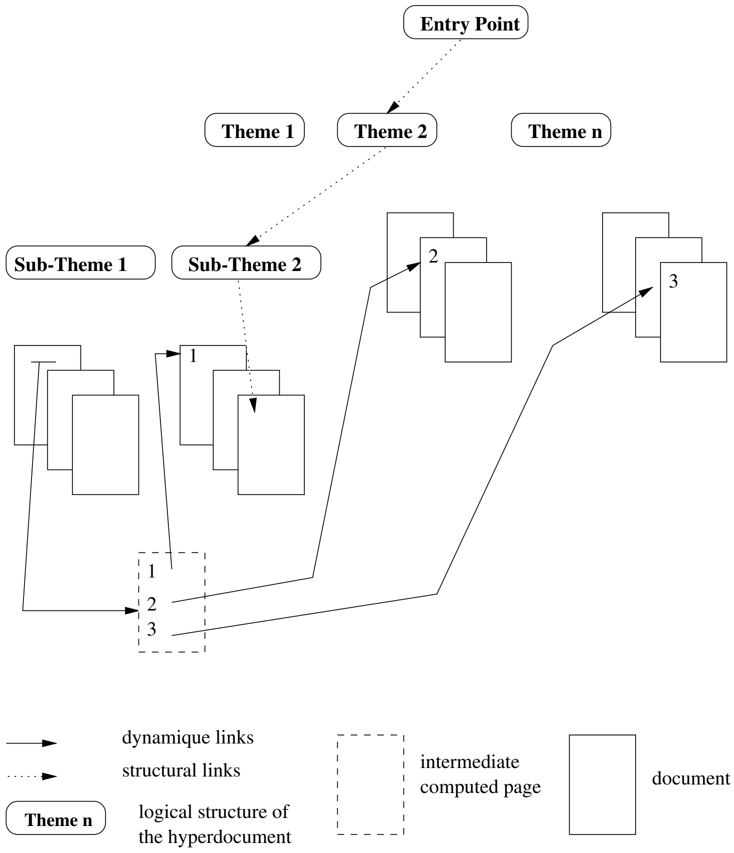

Figure 1: The two ways of browsing across the document

We have now a corpus of documents, each of them being labeled by their authors with a list of descriptors and placed in a subtheme of the tree skeleton. Each subtheme is in fact a cluster of documents, that makes sense, at least, from the authors point of view. These clusters are obviously not the only possible ones. This is mainly because the theme and subtheme follow a socio-economic division, and we could imagine a partition by geographical positions or by dates.

We have successfully used *A posteriori* clustering methods; firstly, using the content of the pages (and not only the meta keywords). Secondly, using the structure of the documents (giving a heavier weight to a word present in a level 1 title rather than the same word inside a plain paragraph). A third kind of evidence has been tested in our corpus : links between documents defined by our so called references. These links have been used to compute densely linked subset of documents. It appears, hopefully, that subthemes have been retrieved. Actually, a closer observation of the clusters obtained by that method enables us to point out a new subtheme; that theme is in fact orthogonal to the subthemes defined in the tree skeleton.

This *a priori* structuring method gave good result with little effort asked to the editor and the author, and giving to readers greater facilities.

3 Browsing paths based on clusters

We move now to the browsing of pages already designed. In this context, we can not handle the formatting of such pages and *a priori* approaches (described in the previous section) are not adapted. However, a significant amount of data and information resides in such loosely structured pages collections and it would be a pity to ignore it.

The concept of text and web mining has been introduced in order to use these resources in data mining driven decision making. In this section, we show how some concepts and methods of data mining can be reused and updated in order to discover browsing paths.

3.1 Clusters as a tool to support the navigation

In order to allow relevant paths from pages already addressed by the web, we suggest to structure pages of a same domain in clusters. A cluster gathers pages having related contents. These pages are described by means of features [3] (for example keywords) extracted from pages. According to the number of keywords and their frequencies in the pages, the definition of a cluster can be more or less specific (and a cluster can collect a few or a large amount of pages). The idea is to build a system which is composed of an off-line module that performs cluster analysis and an online module which is responsible for dynamic link generation of web pages. From our practical experience, we think that this strategy is feasible only for pages belonging to a same domain or gathered by editors.

The sound idea to define browsing paths is that a cluster of pages offers a natural way to navigate around a same delimited topic. A cluster is used to determine which pages should be directly linked. When a user visualizes a page, three main options, corresponding to three kinds of browsing, are available:

[3] Features or descriptors are also called *attributes* in the machine learning community.

- the user is strongly interested in the current page: to pursue the course, we propose pages coming from the same cluster. In other words, they are the most similar pages relatively to the descriptors corresponding to the current cluster.
- the user estimates that the content of the current page covers a too specific area. It is needed to relax constraints about the descriptors of the cluster. Ignoring some of theses descriptors points out more general clusters and their pages are suggested. This way allows to highlight additional links to lead users to close or new topics.
- the current page is estimated as too general. A reverse process to the one described below is applied: pages associated to more specific clusters are presented in order to focus the topics of the browsing.

For the time being, descriptors characterizing pages are essentially based on keywords extracted from the text in the pages. This means that, in order to make use of such kind of descriptors, the text in the pages must first be preprocessed. It would be also interesting to take into account the graphics and other multimedia files. Multimedia clustering is very resource intensive and still a relatively new topic.

3.2 Clusters based on association

3.2.1 Association rules and clustering

The production of meaningful clusters in data mining is a hot topic. We focus here on association rules which is one of the most popular methods and we show here their potential contribution in the discovery of clusters.

An association rule [AMSTI96] is a statement of the form `''95% of pages that have boat and passengers get harbor''`, 27% of pages in a hyper-document match this rule. This last number is called *support* of the rule, 95% is the *confidence* (i.e. the percentage of data that contain the consequent among those containing the antecedent). `boat` is an item (here, a keyword) and both antecedent and consequent are sets of items (or *itemsets*). The question of mining rules that satisfy a user-specified minimum support and minimum confidence has been extensively studied and fast algorithms exist [AMSTI96] [TOI96] allowing to explore quickly very large databases with a minimum number of pass on data. Briefly speaking, mining association rules is composed of two steps. The first one consists in finding all frequent itemsets (that means, itemsets with support greater or equal to the minimum support) and the second is in generating from frequent itemsets all rules with a confidence greater or equal to the minimum confidence. We know that, once all frequent itemsets and their support are known, the generation of rules is straightforward. Hence, only the first step has a high computational cost.

There exist two main families of methods of clustering directly stemmed from association rules. The first one consists in grouping pages in order to minimize an intra-cluster and inter-cluster costs [WCL99]. Typically, an intra-cluster cost measures differences between pages within the cluster and an inter-cluster cost indicates the duplication of same items among different clusters. This strategy gives a partition of the examples, but it is not able to derive easily a characterization of each element of the partition.

The second family gathers methods like hypergraph clustering. The latter [HKKM97] takes a set of association rules and declares the items in the rules to be vertices, and the rules themselves to be hyperedges. Since each association rule has a direction, the algorithm combines all rules with the same set of items, and uses an average of the confidence of the individual rules as the weight for a hyperedge. Clusters are then found by a hypergraph partitioning method [KH00]. We do not obtain directly a clustering of examples and these ones must be ranked in a second time into clusters. But, we get a description of the extracted clusters in terms of the items.

3.2.2 Discovering clusters of pages

An itemset brings an intuitive way to define a cluster: it gathers pages who are similar according to items that are making up. A sound idea is to use the itemsets generated during the process of mining the rules We notice that, for an itemset, we have no information about items not found in it. For example, if the itemset does not contain an item relative to the kind of transport, two pages belonging to it can have equally a similar or different kind of transport. This point differs from most usual clustering techniques which use a global measure of similarity between data.

The outline of a method to produce clusters that we have defined is presented now. It starts from the frequent itemsets in order to take into account the "importance" of the association between the items. In our context, this means that we search multi keywords that tend to co-occur in the pages of the collection. In the aim of clustering, for a group of data (here, the pages), we prefer to simply produce the single itemset which is composed of the maximal number of items shared by the group: such an itemset is called *closed* itemset [PBTL99]. The idea is to catch the maximum amount of similarity among the data.

If some exceptions occur between correlations in the data, this means that the "best" rules are not characterized by a confidence level of 100%. This situation is typical from domains where there is some uncertainty intrinsically embedded in the data. In order to integrate this fact in our work, the next step is to mine the *almost* closed frequent itemsets: we relax the constraint of closure which is on the core of the closed itemsets [BB00] to allow few exceptions for a cluster. Furthermore, we offer to the experts to introduce some constraints: these ones can be connected with the domain (as the selection of some items) or linked to the characteristics of the clusters (like the minimum number of items, pages, and so on). It is also a pragmatic way to reduce the number of itemsets. Let us note that we obtain here a set of clusters, which is not inevitably a *clustering* (in the sense of a partition of the data). This set a cluster allows a navigation through pages as we have explained above.

The use of clusters and, more precisely, the structuring of a set of pages in clusters is not new. For instance, Godin and al. [GMA93] have designed a user interface which permits enlargement or refinement of the user's query by navigating through a graph of keywords and associated pages. This graph is a Galois lattice structure. Each vertex represents a query formed of a conjunction of keywords with the retrieved documents. If we relate our approach to Godin's work, we introduce the notion of *frequent* (for a cluster) and the relaxation of the closure constraint. We think that these notions are required when the domain becomes large and it is in general the situation in web mining.

4 Conclusion

The definition of interesting paths through a hyper-document is often subjective and based also on the domain knowledge of the user. Authoring tools are very useful to index and design a hyper-document which will offer several browsing paths recognized by the authors. A limit of this approach is that it is not able to manage pages already designed. In this last situation, we have seen the interest and the role of clusters for the navigation.

Another contribution of this strategy is to allow a navigation more driven by the own interests of the surfer. He can choose if he prefers to enlarge or refine the content of the pages and on which topics. Furthermore, casual users without any prior knowledge of the pages or users without any particular subject in mind can freely navigate. If we have a look about information retrieval area, let us notice that we can combine clustering of documents and queries. We think that users prefer queries for subject retrieval, when he has already an idea of the expected information, and intuitive and suggested navigation for browsing. It is quite hard to ask relevant queries faced with unknown documents. In such situations, we claim that clusters of pages are a useful way for browsing. Furthermore, although indexing and selecting keywords are well-studied within information retrieval, we think that associations of keywords may be a way to improve the quality of retrieval and access to text and web data.

Finally, let us note that it is hard to evaluate quantitatively the quality of the discovered clusters because, unlike classification, there is no predefined "correct value" to use to calculate measures such as precision or recall. Nevertheless, in our situation, we have the opportunity to have corpus containing keywords and browsing paths established by authors: it is the corpus provided by the authoring tool OhRAGE. We plan to use this corpus as a test set: we will confront browsing paths corresponding to clusters obtained by associations with those specified by the editorial board of the document.

References

AMSTI96 Agrawal, R. and Mannila, H. and Srikant, R. and Toivonen, H. and Verkamo, A. I., Fast discovery of association rules, Advances in Knowledge Discovery and Data Mining, 12, AAAI/MIT Press, 1996

BB00, Boulicaut, J. F. and Bykowski, A., Frequent closures as a concise representation for binary data mining, proceedings of the Fourth Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 00, Lecture notes in artificial intelligence, 1805, Springer-Verlag, Kyoto (Japan), 62-73, 2000.

CH97 C. Chang and C. Hsu. Customizable multi-engine search tool with clustering. In Proc. of 6th International World Wide Web Conference, 1997.

CMS97 Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava, Web Mining: Information and Pattern Discovery on the World Wide Web (A Survey Paper) (1997), in Proceedings of the 9th IEEE International Conference on

Tools with Artificial Intelligence (ICTAI'97), November 1997.

GMA93, Godin, R. and Missaoui, R. and April, A., Experimental comparison of nsavigation in a Galois lattice with conventional information retrieval methods, nt. J. Man-Machine Studies, 38, 747-767, 1993.

HKKM97, Han, E-H. S. and Karypis, G. and Kumar, V. and Mobasher, B., Clustering based on association rule hypergraphs, proceedings of the workshop on Research Issues on Data Mining And Knowledge Discovery, SIGMOD 97, 1997.

KH00 Karypis, G. and Han, E. H. S., Concept indexing: a fast dimensionality reduction algorithm with applications to document retrieval and categorization, University of Minnesota, TR-00-0016, 2000.

PA97 P. Merialdo P. Atzeni, G. Mecca. Semistructured and structured data in the web: Going back and forth. In Proceedings of the Workshop on the Management of Semistructured Data (in conjunction with ACM SIGMOD), 1997.

PBTL99 Pasquier, N. and Bastide, Y. and Taouil, R. Lakhal, L., Discovering frequent closed itemsets for association rules, proceedings of 7th International Conference on Database Theory ICDT 99, Lecture notes in artificial intelligence, 1331, Springer Verlag, Jerusalem (Israel), 299-312, 1999.

SPE97 E. Spertus. Parasite: mining structural information on the web. In Proc. of 6th International World Wide Web Conference, 1997.

TOI96 Toivonen, H., Sampling large databases for association rules, proceedings of the 22nd International Conference on Very Large Databases, VLDB 96, India, 134-145, 1996.

WCL99 Wang, K. and C, X. and Liu, B., Clustering transactions using large items, proceedings of ACM CIKM 99, 1999.

WVS96 R. Weiss, B. Velez, M. A. Sheldon, C. Namprempre, P. Szilagyi, A. Duda, and D. K. Gifford. Hypursuit: a hierarchical network search engine that exploits content-link hpertexxt clustering. In Hypertext'96: The Seventh ACM Conference on Hypertext, 1996.